



UNIVERSITAT_{DE}
BARCELONA

PROPUESTA DE PLANEACIÓN PARA LA CREACIÓN DE UNA NUEVA VERSIÓN DE LA APLICACIÓN PARA TELÉFONOS INTELIGENTES “AMB MOBILITAT”

Autor: Alfredo Jordán Murillo

Tutor: Juan José Boté

Máster en Gestión de Contenidos Digitales

Junio 2019

Tabla de Contenido

Resumen Ejecutivo	1
1.1. Descripción del Proyecto	1
1.2. Modelo de Negocio y Cadena de Valor	1
1.3. Mercado Objetivo y Potencial	2
1.3.1. Mercado Objetivo	2
1.3.2. Mercado Potencial	2
1.4. Estado Actual del Proyecto	3
1.4.1. Resumen de Resultados del Test de Usabilidad	3
1.4.2. Oportunidades de Mejoras y Prototipo	7
1.5. Equipo de Trabajo	9
2. Reto de Negocio	9
2.1. Business Case	9
2.1.1. Beneficios	9
2.1.2. Costes	10
2.1.3. Riesgos Principales.....	10
2.1.4. Calendario	11
2.1.5. Opciones de Negocio	11
2.1.6. Evaluación de la Inversión	13
2.2. Objetivos	13
2.2.1. Objetivo General	13
2.2.2. Objetivos Específicos.....	13
3. Metodología Aplicada	14
3.1. Camino de Descubrimiento	15
3.2. Camino de Desarrollo.....	17
3.2.1. Dueño del Producto.....	18
3.2.2. ScrumMaster	18
3.2.3. Equipo de Desarrollo	18

3.2.4.	Eventos de Scrum	19
3.2.5.	Otros Términos de Scrum.....	19
4.	Marco Referencial	20
4.1.	Benchmarking.....	20
4.2.	Procesos Relacionados	24
4.3.	Casos de Uso.....	25
4.4.	Stack Tecnológico.....	26
4.4.1.	Tecnologías Client-side	26
4.4.2.	Tecnologías Server-side	27
4.5.	Datos	28
4.5.1.	Leyes Aplicables.....	28
4.5.2.	Recolección de Datos	29
4.5.3.	Explotación	30
4.6.	Riesgos del Proyecto	30
4.6.1.	Riesgos Tecnológicos.....	30
4.6.2.	Riesgos Económicos	31
4.6.3.	Riesgos de la Metodología	31
5.	Planificación	32
5.1.	Cronograma.....	32
5.2.	Plan de Trabajo.....	33
5.3.	Recursos Necesarios	35
5.3.1.	Recursos Humanos.....	35
5.3.2.	Recursos Materiales no Tecnológicos	36
5.3.3.	Recursos de Hardware.....	36
5.3.4.	Recursos Software.....	37
5.4.	Indicadores de Rendimiento.....	38
5.4.1.	Métrica de Entregables	38
5.4.2.	Métricas de Efectividad	39
5.4.3.	Métricas de Equipo de Trabajo	39
6.	Presupuesto.....	39

7.	Consideraciones.....	40
7.1.	Recomendaciones Iniciación y Setup.....	40
7.2.	Consideraciones Sprints de Descubrimiento.....	41
7.2.1.	Design Sprint	41
7.2.2.	Google Material Design	41
7.2.3.	UI, UX y Prototipo	43
7.3.	Consideraciones Sprints de Desarrollo	43
8.	Política de Difusión	44
8.1.	Objetivo	44
8.2.	Estrategia	44
8.3.	Plan de acción	45
9.	Fuentes Documentales Consultadas.....	45
10.	Anexos	46

Resumen Ejecutivo

1.1. Descripción del Proyecto

El proyecto propuesto es el rediseño de la interfaz gráfica y funcionalidades de la aplicación AMB Mobilitat por medio de la proposición de una metodología ágil, ofreciendo un prototipo de alta calidad como punto de partida. Este prototipo está basado en el Sistema de Diseño Material Design de Google, además de tendencias y funcionalidades propias del mercado de aplicaciones de movilidad. El proyecto nace de las oportunidades de mejoras de UI y UX, evidenciadas en los resultados de un test de usabilidad.

Se propone específicamente cambiar la estructura y funcionalidades para poder ofrecer una experiencia, con cierto nivel de accesibilidad, que satisfaga las necesidades de transportación de los usuarios y de esta manera fidelizarlos. Además, a largo plazo, permitirá competir con aplicaciones líderes de este mercado tales como Google Maps, Moovit, CityMapper entre otras.

Además estos cambios propuestos permitirán establecer un Sistema de Diseño propio, útil para las demás aplicaciones gestionadas por AMB creando una imagen de marca mucho más robusta.

1.2. Modelo de Negocio y Cadena de Valor

Se propone mantener la aplicación gratuita, sin publicidad de ningún tipo. La estrategia se basará en expandir la base de usuarios y hacer branded content.

Esto permitirá sentir a los usuarios que tienen una aplicación que aporta valor y estarán más motivados a dejar buenas calificaciones y como consecuencia subir de rating en las tiendas virtuales correspondientes. A mediano plazo se esperaría competir con los líderes de nuestro mercado objetivo.

La amortización del coste de desarrollo inicial y mantenimiento continuo se lo propone hacer por medio de los presupuestos de AMB, ya que a largo plazo cuando la imagen de marca digital sea positiva, captará usuarios de otros medios de transportes públicos que usen aplicaciones para dispositivos móviles.

1.3. Mercado Objetivo y Potencial

1.3.1. Mercado Objetivo

Al ser este un grupo potencial de consumidores, es esencial que los identifiquemos para poder enfocar los recursos eficientemente. Las características demográficas de nuestro público son las siguientes:

- Edad: 16 - 64 años
- Sexo: Femenino y masculino
- Ubicación: Área Metropolitana de Barcelona
- Nivel de ingresos: Clase media y alta
- Nivel de educación: Bachillerato y universitario
- Estado civil: Indiferente
- Ocupación: Estudiantes, trabajadores

Según los datos estadísticos de IDESCAT¹ al año 2018, esto nos da un resultado de 1.488.699 personas.

1.3.2. Mercado Potencial

Se considera como mercado potencial a todas las personas no locales, en su mayoría turistas, que visitan el área metropolitana de Barcelona por un tiempo determinado.

Las características demográficas del grupo anteriormente mencionado son:

- Edad: 18 - 64 años
- Sexo: Femenino y masculino
- Ubicación: Fuera del Área Metropolitana de Barcelona
- Nivel de ingresos: Clase Media y Alta
- Nivel de educación: Bachillerato y universitario
- Estado civil: Indiferente
- Ocupación: Estudiantes, trabajadores viajeros y jubilados

De acuerdo a este perfil y las estadísticas de turismo ofrecidas por el Ayuntamiento de Barcelona² en el año 2017, el tamaño del mercado potencial es de 8.884.550 personas.

¹ IDESCAT, el municipio en cifras <https://www.idescat.cat/emex/?id=13&lang=es#h1fffff>

² Informe de l'activitat turística a Barcelona 2017
https://ajuntament.barcelona.cat/turisme/sites/default/files/informe_act_tu_2017_complet_0.pdf

1.4. Estado Actual del Proyecto

Como se mencionó anteriormente, el proyecto parte de los resultados de un test de usabilidad presencial moderado (ficha técnica en el siguiente punto), realizado a tres habitantes de Barcelona cuyo objeto de estudio fue la aplicación actual de AMB Mobilitat. Según Jacob Nielsen, para obtener beneficios de un test de usuarios³ es necesario realizar el test a 3-6 personas.

El sitio web referente en usabilidad Usability.org (2004) define los test de usuario como:

“La evaluación de un producto o servicio mediante la prueba con usuarios representativos. Normalmente, durante una prueba, los participantes intentarán completar las tareas típicas mientras los observadores observan, escuchan y toman notas. El objetivo es identificar cualquier problema de usabilidad, recopilar datos cualitativos y cuantitativos y determinar la satisfacción del participante con el producto.”

El test de usabilidad es una herramienta que permite generar informes para rediseñar una web o aplicación además validar o desmentir suposiciones en cualquier momento del proceso de desarrollo, desde wireframes a páginas webs o aplicaciones en producción.

1.4.1. Resumen de Resultados del Test de Usabilidad

Número de usuarios y características:	El test se realizó a 3 usuarios: <ul style="list-style-type: none">• Residentes españoles• Diversas edades y ocupaciones• Parentesco familiar
Fecha de las sesiones:	<ul style="list-style-type: none">• Sábado, 5 de enero de 2019
Dispositivos usados:	<ul style="list-style-type: none">• 2 laptops (Windows OS, Morae Techsmith)• Un móvil Samsung S9 Android versión 9 con la aplicación AMB versión 4.2.0
Técnica empleada:	<ul style="list-style-type: none">• Test de usabilidad moderado presencial/Cuestionarios

³ How Many Test Users in a Usability Study? <https://www.nngroup.com/articles/how-many-test-users/>

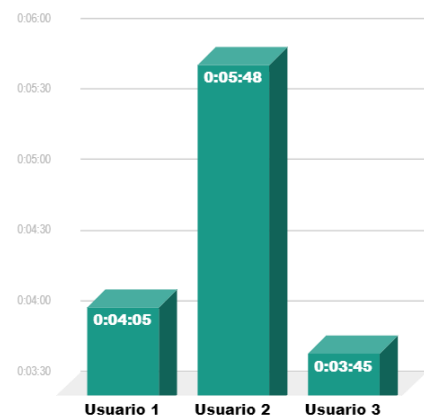
Al inicio del test se realizó una pequeña encuesta para conocer las aplicaciones usadas por los participantes, estos son los resultados:

- 2 usuarios Google Maps, 1 AMB Mobilitat para transporte público
- Piensan que Google Maps contiene más información sobre todos los medios de transporte con relación a AMB Mobilitat.

El test consistió en la realización de tres tareas, a continuación los resultados de cada una de ellas:

Tarea 1: Evaluar si el usuario puede encontrar rápidamente transporte del punto A al punto B

Pregunta: Imagina que el día de mañana necesitas transportarte del CC Diagonal Mar al Arco del Triunfo, debido a que te vas a encontrar con un grupo de amigos a la 1pm. Usa la aplicación AMB Mobilitat para averiguar cuál es el medio de transporte y ruta más óptima para llegar a tu destinación a la hora pactada. ¿Cómo lo harías?



Observaciones:

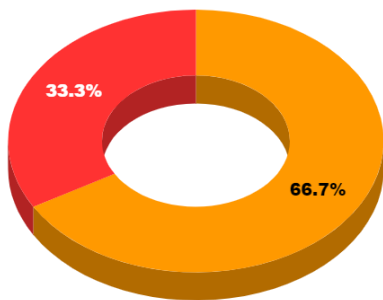
- El problema recae en el buscador de la sección “¿Dónde quieres ir?”, debido a que los 3 participantes quisieron ingresar el nombre de las atracciones en los input text “De:/A:” y no las direcciones exactas, debido a que los usuarios ya tienen establecido el modelo mental de la función de Google Maps.
- La aplicación no tiene ningún tipo de feedback para guiar a los usuarios sobre los valores aceptados por el buscador, solamente un diálogo con opciones en inglés “Current location” y “Location on map”, cabe recalcar que en configuraciones estaba

seleccionado el idioma castellano.

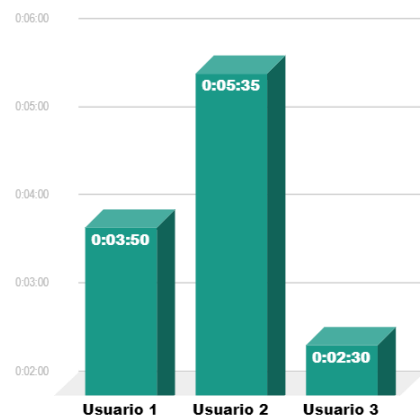
- Todos los usuarios utilizaron la opción "Location on map" al no tener respuesta alguna con el nombre de las ubicaciones. Sólo 2 pudieron seleccionar las direcciones. Una vez más, el modelo mental creado por el uso de Google Maps afectó la UX, debido a que al seleccionar la ubicación en el mapa aparecía un punto más la dirección del mismo pero solo 2 usuarios comprendieron, después de algunos intentos, que se debe hacer clic en la dirección para poder continuar, lamentablemente la aplicación no ofrece ningún tipo de guía o botón de continuar/seleccionar, más allá del nombre de la sección "Seleccionar origen/destino". Cabe mencionar que los usuarios son ciudadanos de Barcelona, esto con un turista que desconoce la ciudad no funciona.
- Una vez que la aplicación mostró los resultados a los dos usuarios que llegaron a este punto, les resultó fácil poder cambiar la hora y cambiar los medios de transporte, es importante mencionar que estas dos últimas funcionalidades les gustaron.

Tarea 2: Evaluar si el usuario puede agregar una parada de bus a sus favoritos

Pregunta: Ahora, imagina que deseas guardar una parada como favorita. ¿Cómo lo harías?



● Completado con dificultad ● No completado



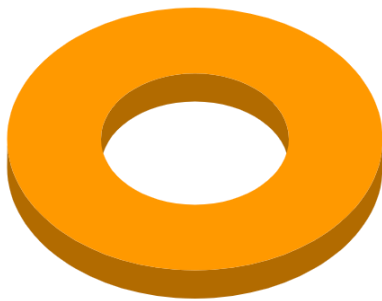
Observaciones:

- Un usuario mencionó sentirse "perdido", al confundir el buscador interno de la sección de favoritos como buscador de todas las paradas, podemos decir que se debe a la falta de información/guía sobre cómo realizar la tarea en la sección de favoritos.

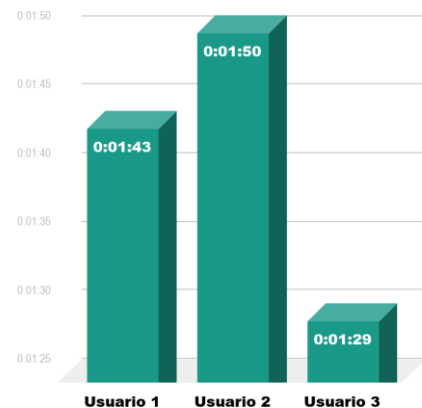
- A pesar que la forma más fácil de completar esta tarea era desplegar la información de cualquier parada y darle clic a la estrella en la parte superior izquierda, ninguno de los 2 usuarios que completaron la tarea lo hicieron de esta manera. Optaron por el camino largo que es agregar la parada por el código de la parada desde el apartado de "Favoritos", lo cual significa en buscar la parada desplegar la información, recordar el nombre, ir al apartado de "Favoritos", dar clic en el botón "+" e ingresar el código.

Tarea 3: Evaluar si el usuario puede descargar a su móvil el recorrido y horario de una línea de bus

Pregunta: Supongamos que usas frecuentemente la línea "B20" y deseas más información sobre su recorrido y horarios, por lo tanto deseas descargar la ficha informativa. ¿Cómo lo harías?



● Completado con dificultad



Observaciones:

- La aplicación tuvo problemas para realizar la descarga, la conexión con el servidor termina antes de terminar la descarga.
- Un dato curioso durante esta prueba de usuario, es que la aplicación cambio el idioma a inglés de manera autónoma, esto impresionó negativamente al usuario.

Conclusiones generales del test de usabilidad:

- La aplicación tiene mucha ventaja potencial sobre Google Maps, AMB crea/gestiona la información, pero su usabilidad e interfaz gráfica podrían

perjudicar su éxito. A pesar de estos problemas los usuarios están predispuestos a seguir usándola en el futuro.

- Hay que adaptarse a los modelos mentales de los usuarios, los cuales son formados por las aplicaciones predominantes del mercado, en este caso Google Maps.

1.4.2. Oportunidades de Mejoras y Prototipo

Como resultado de los test de usuarios, se obtuvieron varias oportunidades de mejoras a nivel de interfaz y experiencia de usuario:

- **Búsqueda de itinerarios**

- Mantener el nombre de la sección *¿Dónde quieres ir?* para evitar confusiones y disminuir la carga cognitiva.
- Permitir que los usuarios puedan ingresar nombre de locaciones conocidas como centros comerciales, atracciones turísticas, bares populares, etc. Un bono extra es que ayudaría a que turistas usen más la aplicación. Otra opción más rápida de implementar es mostrar un mensaje de error al usuario pidiéndole que ingrese una dirección, ejemplo *"Por favor, ingresa una dirección completa Ejemplo: Avenida Diagonal, 800"*.
- Se recomienda que se elimine el popup de filtro de transporte y funcione como dropdown junto a un label *"Transporte"* y el campo por default sea *"Todos"*.
- Cuando el usuario todavía no haya llenado los campos *"De/A"* mostrar un mensaje de ayuda, ejemplo *"Ingresa las direcciones para ver los itinerarios de transporte"*.
- Añadir texto *"Itinerarios: XX"* para indicar la cantidad de itinerarios mostrados.
- Agregar una opción para organizar la lista de resultados *"Menor a mayor tiempo, menos tiempo de caminata, menos transbordos, etc"*.
- Cambiar Texto *"A"* por *"De"*, lo mismo aplica para Texto *"B"* por *"A"*.

- **Favoritos**

- Cambiar de posición/funcionalidad el botón de agregar paradas favoritas a un botón flotante para que sea más visible.
- Posibilidad de eliminar este buscador interno ya que los usuarios lo confundieron como buscador de líneas.
- Agregar una opción para organizar/agrupar la lista de paradas favoritas
- Adicionar al texto actual, una pequeña explicación de cómo agregar una parada favorita.

- Posibilidad de agregar paradas por el nombre no solo por código, además de adicionar los botones de cancelar y aceptar en el popup de agregar parada. No hay mensaje de error cuando el usuario ingresa un código incorrecto, no hay ningún tipo de retroalimentación.
 - La mayoría de los usuarios les gustaría que no solo sean guardadas paradas sino también líneas de medios de transporte e itinerarios.
 - Mejorar la estructura de la ficha informativa de la parada, los elementos están muy agrupados, usar más espacio blanco para permitir una lectura más fácil. Ninguno de los usuarios pudo darse cuenta del Call to Action (CTA) de la estrella (agregar a favorito).
- **Listado de líneas**
 - Cambiar el ícono de inicio de esta sección, confunde a los usuarios ya que da a entender que solo se puede encontrar líneas de buses, cuando se puede encontrar líneas de todos los sistemas de transporte. En la versión de inglés la sección se llama “List of bus lines” lo cual contaría como error de contexto también.
 - Agregar una opción para organizar/agrupar la lista de líneas, ejemplo por medio de transporte.
 - Mejorar la estructura de la ficha informativa de la línea, los elementos están muy agrupados, usar más espacio blanco para permitir una lectura más fácil.
 - Asegurarse que todas las líneas tengan sus fichas informativas o deshabilitar la descarga en caso de no tener su respectivo archivo pdf.
 - Aumentar el tamaño vertical del iframe donde se ve la descarga y se adicione un botón con el texto “Descargar a mi dispositivo”.
- **Otras recomendaciones**
 - Implementar un tutorial la primera vez que los usuarios ingresen a la aplicación
 - No dejar que el idioma se cambie sin que lo haya hecho el usuario, verificar que los textos coinciden en todos los idiomas
 - No usar mayúsculas en las fichas informativas para mejorar la legibilidad (accesibilidad)
 - Mostrar a los usuarios mensajes de error en los campos de ingreso

Para ver el prototipo aplicado con algunas de estas mejoras en versión Android, referirse al anexo 1.

1.5. Equipo de Trabajo

La metodología que se plantea para este proyecto es un framework de Agile, llamado Dual-Track Scrum, que mezcla trabajo de desarrollo de código y UX. Para este framework es preferible trabajar con un equipo de trabajo pequeño (5-9 personas), según un estudio realizado por Scrum alliance⁴. Por lo tanto, se propone los siguientes roles:

- Scrum Master
- Cliente(Product Owner)
- Equipo de trabajo
 - 2 diseñadores UI/UX senior
 - 1 desarrollador full stack aplicaciones senior
 - 2 desarrolladores full stack aplicaciones junior
 - 1 quality assurance

La composición del equipo podrá cambiar una vez se haya acordado el alcance de la implementación final.

2. Reto de Negocio

2.1. Business Case

2.1.1. Beneficios

En la actualidad es muy importante que los productos digitales de las empresas cumplan con cierto grado de calidad, que se ve reflejado en las calificaciones y comentarios online, lo cual les permite crear una buena imagen de la marca⁵.

Entre los principales beneficios que tendría la propuesta de un rediseño de la aplicación AMB mobilitat, son los siguientes:

- Captación y fidelización de usuarios
- Impulsar reconocimiento de marca a nivel local e internacional (turistas)

⁴ State of Scrum 2017-2018

[https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2017-SoSR-Final-Version-\(Pages\).pdf](https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2017-SoSR-Final-Version-(Pages).pdf)

⁵ Usability Testing <https://www.usability.gov/how-to-and-tools/methods/usability-testing.html>

- Sobresalir en el mercado de aplicaciones Android y iOS de transportación pública del área metropolitana de Barcelona.
- Visibilizar y facilitar acceso a otros servicios de transportación pública (bicing, avanzar, drivy, etc).
- Implementación de criterios básicos de accesibilidad móvil
- Crear una guía de diseño y referente para las demás aplicaciones de AMB

2.1.2. Costes

El coste estimado para el proyecto está dividido en los siguientes rubros:

Concepto	Importe (€)
RRHH	86.900,00 €
Profesionales Externos	8.000,00 €
Administración	7.020,00 €
Software/Hardware	2.880,00 €
Total Neto =	129.952,00 €

2.1.3. Riesgos Principales

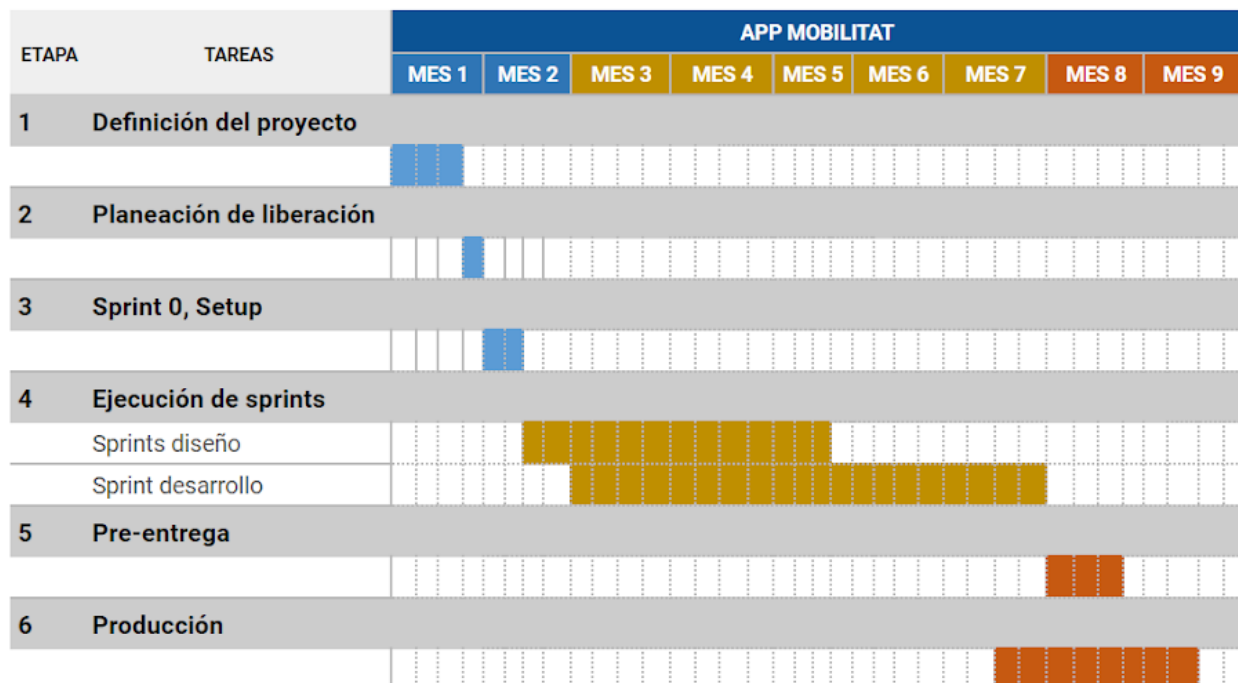
Los riesgos posibles para los proyectos digitales de esta naturaleza son los siguientes:

- **Riesgos tecnológicos**
 - No poder implementar funcionalidades planteadas en el prototipo por problemas de compatibilidad con el sistema de entrega de información actual.
 - Elementos del stack tecnológico dejen de tener soporte y la migración resulte en un proceso complejo y costoso.
 - Tener que depender de terceros para la prestación de servicios indispensables.
 - Tener que crear diferentes versiones de la aplicación debido a que ciertas funcionalidades no sean compatibles en dispositivos de gama baja.
- **Riesgos económicos**
 - Que los usuarios no se interesen por usar la aplicación y opten por aplicaciones más populares y la inversión no logre alcanzar los beneficios planteados anteriormente.

- Que la aplicación sea muy popular entre los usuarios actuales del transporte público, y los costos de servicios de la API de mapas incrementen afectando el presupuesto proyectado.
- **Riesgos de la metodología**
 - Estimación de tiempos muy optimista
 - Cambios en el equipo por razones externas
 - Cambios o definición no apropiada de criterios de aceptación del sprint
 - Aprendizaje de Skills nuevos
 - Dinámicas negativas en el equipo de trabajo

2.1.4. Calendario

El tiempo estimado del proyecto es de 8 meses y medio, desglosados en los siguientes módulos:



2.1.5. Opciones de Negocio

Otras opciones consideradas:

- **Mantener la interfaz gráfica y experiencia de usuario de la aplicación actual**

La aplicación actual, como se demostró en el estudio previo, no ofrece una experiencia de usuario satisfactoria. Esto también se ve reflejado, al mes de abril del 2019, en las calificaciones de las tiendas App Store (2,2/5) y Google Play (3,7/5). Se podrían sumar

a esto los comentarios negativos de las propias tiendas virtuales que tienden a generar una mala imagen alrededor del producto.

Hay que recordar que el comportamiento general de los usuarios antes de adquirir un producto digital que va a consumir recursos de su dispositivo móvil, es tomar como referencia la calificación y los comentarios de otros usuarios. Estas referencias tienen un poder decisión del 68% sobre referencias de amigos o familiares⁶.

Por lo tanto, si se decide mantener la aplicación actual, no habrá beneficios. Habrá usuarios que opten por usar otras aplicaciones y al mismo tiempo habrá una connotación negativa alrededor de la marca, que podría afectar a los otros productos digitales de AMB.

- **Plantear una aplicación que solo involucre cambiar una mínima parte de la interfaz gráfica y experiencia de usuario**

Se considera esta opción, por si se decide implementar parte del proyecto. Los costos y el tiempo de desarrollo son menores y en cierto grado se mejorará el *look and feel* de la aplicación pero manteniendo las mismas funcionalidades.

En medida de lo que permite el framework de desarrollo actual, se aplicarían los estándares de UI y UX establecidos en la guía Material Design.

Por lo tanto, si esta opción es seleccionada, se debe tomar en cuenta que los usuarios que tienen instalada la aplicación en un primer vistazo, notarán cambios y mejoras a cierto grado pero podrán sentirse decepcionados debido a que las prestaciones de fondo siguen siendo las mismas. Esto último se podría ver reflejado en las calificaciones y comentarios de las tiendas virtuales correspondientes.

- **Plantear una aplicación en el cual la interfaz y las funcionalidades del flujo de solicitud de itinerarios son mejoradas**

En esta opción se toma como referencia la funcionalidad principal de la aplicación, la cual facilita itinerarios de movilidad en transporte público, para replantear por completo a nivel de diseño y funcionalidades tomando en cuenta las limitaciones del framework

⁶ Most common purchase influences of online shoppers <https://www.statista.com/statistics/788185/us-online-shoppers-purchase-influences/>

actual. Además se aplicarían mejoras a nivel global de UI y UX propuestas en la opción anterior.

La diferencia primordial con la propuesta anterior es que en este caso, al momento de llevar el prototipo a producción se necesitaría un esfuerzo mayor del lado de programación y por ende un tiempo y costos de desarrollo mayores.

Es la opción que más beneficios ofrecería ya que la experiencia de usuario mejoraría considerablemente solo modificando el flujo principal, debido a que los usuarios se llevaría una buena impresión de la aplicación, siempre y cuando sus expectativas básicas (itinerarios de movilidad) sean cumplidas.

2.1.6. Evaluación de la Inversión

Tomando en cuenta que la aplicación no genera ingresos monetarios, se propone evaluar el Valor de la Inversión o VOI (inglés) enfocado en satisfacción del usuario, reconocimiento de marca y fidelización. El VOI se puede reflejar mejor en los beneficios planteados anteriormente.

Estos valores intangibles, a la larga significan ahorros en áreas como atención al cliente debido a comentarios y quejas por parte de los usuarios mediante el teléfono o bien en las diversas redes sociales.

2.2. Objetivos

2.2.1. Objetivo General

El fin de este proyecto es:

- Ofrecer una planificación basada en metodologías ágiles para el desarrollo de una aplicación para dispositivos Android y iOS, que satisfaga las necesidades de transportación de los usuarios, por medio de estándares modernos de UI y UX planteados por la guía de diseño Material Design de Google.

2.2.2. Objetivos Específicos

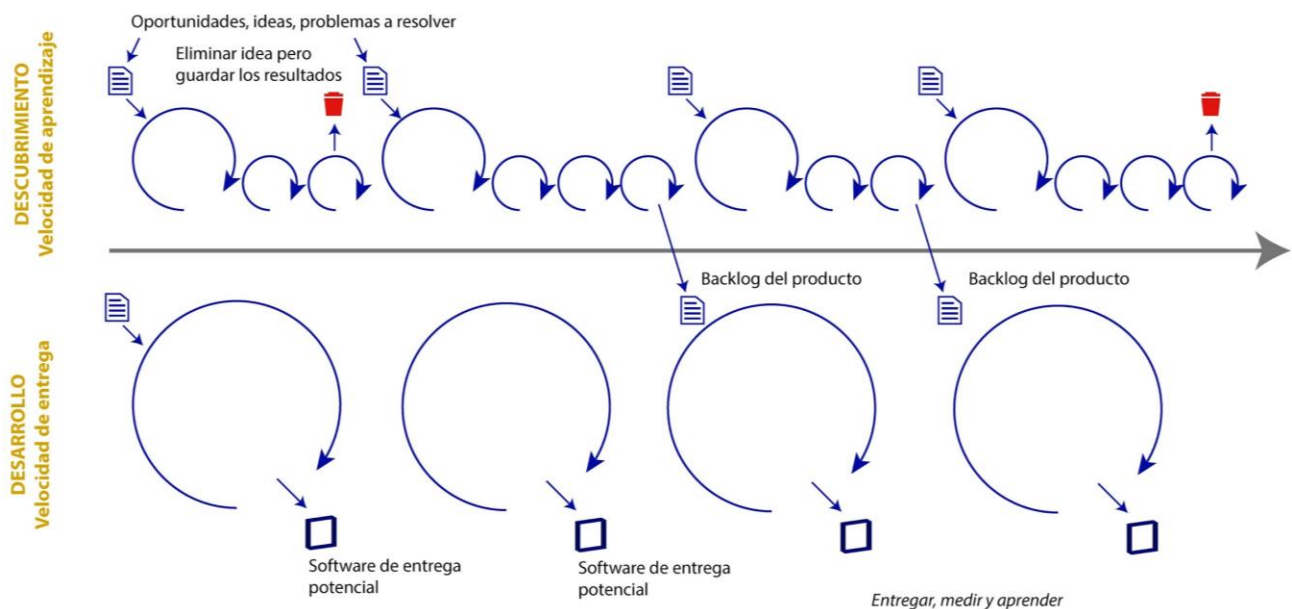
Se logrará el objetivo anteriormente planteado a través de:

- Ofrecer una nueva metodología para mejoras futuras (Dual-Track Scrum)
- Potenciar y definir la funcionalidad principal (búsqueda de itinerarios)

- Potenciar y definir nuevas funcionalidades secundarias (búsqueda de líneas, agregar favoritos y notificación de alertas de líneas de transporte), con el fin de ofrecer una aplicación más robusta.
- Aplicar conceptos de accesibilidad básicos en la aplicación gracias a la guía de Material Design.
- Proponer implementación de una Guía de Diseño de AMB para iOS y Android
- Proponer la recolección de datos para UX
- Proponer un plan básico de marketing online y offline

3. Metodología Aplicada

La metodología que se utilizará para llevar a cabo este proyecto es un framework de Agile llamado Dual-Track Scrum⁷, propuesto por Marty Cagan y Jeff Patton. Esta metodología permite incluir dentro del Scrum tradicional tareas enfocadas al diseño y UX (Design Sprints).



Flujo completo de metodología Dual-Track Development - Jeff Patton

Esta metodología divide la actividades de un equipo de trabajo en 2 caminos paralelos que son descubrimiento y desarrollo, no se debe pensar que son dos procesos diferentes, sino dos partes de un solo proceso. Todo el equipo debe estar involucrado

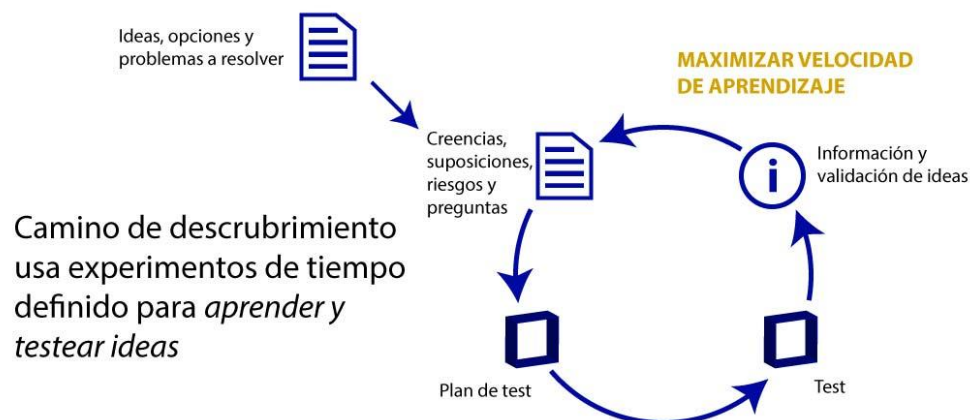
⁷ Dual Track Development is not Duel Track <https://www.jpattonassociates.com/author/jeff/page/2/>

en los dos caminos y deben estar enfocados en el valor del producto entregado y no solo en la rapidez de entrega.

Jeff Patton señala que las características generales de Dual-Track Scrum son:

- El trabajo de desarrollo se centra en la previsibilidad y la calidad
- El trabajo de descubrimiento se centra en el aprendizaje rápido y la validación
- El descubrimiento y el desarrollo se visualizan en dos pistas porque son dos tipos de trabajo y dos tipos de pensamiento diferentes.
- El descubrimiento es una parte necesaria del desarrollo del producto que se puede aplicar con principios ágiles.
- Cambiamos sustanciales del producto y eliminación de ideas poco prácticas
- Se involucran siempre los dos caminos, siempre deben estar conscientes del trabajo de ambos.
- Constante medición y crecimiento sobre el producto durante las diferentes entregas.

3.1. Camino de Descubrimiento



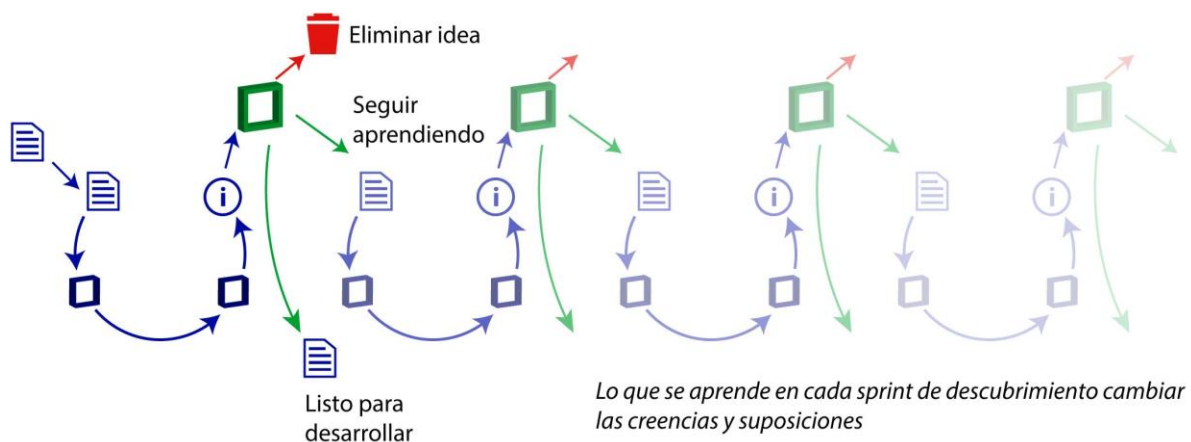
Detalle de un sprint de descubrimiento - Jeff Patton

Este camino basado en Design Sprints 2.0, se enfoca en aprender rápidamente y validar por sprints. En esta etapa están involucrados el dueño del producto, 2 diseñadores UI/UX y 1 desarrollador senior. Generalmente estos sprints pueden durar entre 2 o 4 semanas, 1 semana de análisis y prototipo inicial + 1-3 semanas de iteración del prototipo, dependiendo de la complejidad del problema.

Durante los dos primeros días se comienza por entender y definir el problema, para poder llegar a la mejor solución. Se realizarán reuniones con el equipo de trabajo y el dueño del producto para entender por completo el problema y el contexto de la empresa (limitaciones técnicas, requerimientos del negocio, etc).

Se hallarán varias soluciones al problema a través de técnicas como *brainstorming*, *lightning decision jam*, *prototipos de baja calidad* y *debates*. Una vez hallada la solución adecuada, en los siguientes días se trabajarán con el equipo de diseño y el dueño del producto, se creará uno o varios prototipos de calidad media. Lo importante es conocer si la aplicación/web/producto solucionan el problema planteado, los detalles de UI/UX serán tomados en cuenta en la siguiente etapa de iteración de diseño. Esto permitirá validar ideas con los usuarios finales, a partir de test de usuarios que medirán el éxito del prototipo o si se vuelve a pensar en una nueva solución.

El proceso permite al equipo equivocarse y recuperarse para encontrar la solución que aporte valor, todo mientras los costos son bajos, caso contrario se estaría dirigiendo presupuesto de desarrollo a una idea que está destinada al fracaso. Cabe mencionar que un prototipo puede tener varias semanas de iteración para refinarlo y tener como resultado al final del sprint de diseño un producto listo para el camino de desarrollo.



Varias iteraciones de ciclos de descubrimiento - Jeff Patton

El equipo de diseño debe trabajar de tal manera que después de un número de ciclos pueda crear un Sistema de Diseño para agilizar ambos caminos, el de diseño y desarrollo.

3.2. Camino de Desarrollo



Detalle de un sprint de desarrollo - Jeff Patton

En esta fase es donde el equipo de desarrollo comienza a crear un producto basado en la documentación creada en el camino de descubrimiento. Este camino se puede considerar como Scrum puro. Algunos de los términos usados en esta etapa también pueden ser aplicados al camino de descubrimiento.

Al trabajar con Scrum se obtienen los siguientes beneficios definidos por Scrum.org (2010):

- Obtener versiones de productos más rápido que las metodologías tradicionales como Waterfall.
- Testeo dentro de cada sprint por lo que ofrece la mayor calidad posible
- Permitir que todos los involucrados participen constantemente del proceso de desarrollo, teniendo como resultado transparencia entre la empresa y dueño del producto.
- Identificar los problemas temprano por lo cual se puede responder rápidamente y como consecuencia evitar incrementos inesperados en costes.
- Ofrecer flexibilidad en el proceso, debido a que los cambios son esperados y aceptados.

Para poder comenzar a aplicar esta metodología es necesario indicar los roles de trabajo del Scrum:

- Dueño del producto(Cliente)
- ScrumMaster
- Equipo de desarrollo

3.2.1. Dueño del Producto

Es el responsable de maximizar el valor del proyecto, no es un comité de personas. Sin embargo, puede representar los deseos de un comité, pero solo él toma las decisiones. Algunas de sus responsabilidades son:

- Establecer los requisitos y sus prioridades (backlog del producto)
- Optimizar el valor del trabajo del equipo de desarrollo
- Asegurar que el equipo de desarrollo entienda lo que se desea

El cliente y la empresa desarrolladora deben aceptar las decisiones del dueño del producto, para asegurarse del éxito del proyecto a través de Scrum.

3.2.2. ScrumMaster

Es el responsable de mantener la estructura del framework, es la guía de Scrum (teoría, reglas, valores y prácticas) de todos los involucrados. Es un servidor-líder de los equipos de desarrollo y diseño, ayuda a solucionar obstáculos y a maximizar el valor creado por los mismos. Algunas de sus tareas que debe cumplir:

- Asegurar que los objetivos, alcance y producto sean entendidos por todos los involucrados.
- Promover la autonomía de los miembros del equipo de trabajo

3.2.3. Equipo de Desarrollo

Consiste en un grupo de profesionales de entre 5-9 personas que llevan el proyecto de idea a producto entregable. Es importante recalcar que el equipo de desarrollo es un ente que debe saber manejarse independientemente, ser eficaz y eficiente en sus tareas, también debe tener las siguientes características:

- Ser multifuncionales
- Definir las tareas que pueden ser realizadas en un tiempo determinado

3.2.4. Eventos de Scrum

Una vez definidos estos roles, se debe tener claro los eventos que se llevarán a cabo dentro de un proyecto basado en Scrum:

- **Planificación del Sprint**, el Sprint es un periodo de 1 a 4 semanas, aunque se recomienda usar periodos de 2 semanas para poder evaluar más seguido el avance del proyecto. La planificación del trabajo que se debe realizar en un sprint es llevada a cabo por el ScrumMaster y el equipo de trabajo, en esta reunión se debe convertir los requerimientos dados por el dueño del producto en tareas. Luego se deben definir las tareas que se van a poder completar y quienes van a ser los responsable de hacerlo.
- **Scrum diario**, es una reunión del equipo de trabajo y el ScrumMaster de 15 minutos que se da todos los días a primera hora. En esta reunión se definirá lo que se logró completar el día anterior, lo que se trabajará el día en transcurso, soluciones para problemas que tienen o perciben a futuro cualquiera de los integrantes.
- **Revisión del sprint**, es una reunión que se tiene al final de un sprint para agregar o modificar el backlog del producto. En este evento informal, estarán presentes todos los involucrados incluyendo los inversionistas y se presentará una nueva versión o incremento del producto. El ScrumMaster es el encargado de hacer la presentación del resultado del sprint, de solicitar retroalimentación y responder preguntas. Esta reunión siempre se debe realizar así no haya un producto como tal a presentar, se puede mostrar avances en código o diseño. El dueño del producto es el encargado de dar el visto bueno al resultado del sprint, en caso de que no se apruebe el producto se debe replantear el backlog del producto.
- **Retrospectiva del sprint**, es la reunión entre el ScrumMaster y el equipo de trabajo para evaluar puntos de mejora, lo que ha funcionado o no del proceso. Las mejoras que se obtengan de esta reunión deben ser implementadas en el siguiente sprint.

3.2.5. Otros Términos de Scrum

Otros términos usados dentro de Scrum:

- **Diagrama Burn Down de sprint (por puntos)**, una manera gráfica para determinar el progreso por sprints.
- **Diagrama Burn Down de día (por puntos)**, una manera gráfica para determinar el progreso por días de un sprint específico.

- **Story**, es una descripción corta de una funcionalidad, tomada del backlog del producto, narrada desde el punto de vista del usuario. Sigue el formato de “*Como - usuario específico- yo quiero -objetivo- para -razón específica-* ”.
- **Cubos de complejidad**, número de la secuencia Fibonacci asignado a un story para marcar su complejidad relativa. Van desde el 1 (cubo pequeño) hasta 21 (cubo épico). Por lo general los stories que son asignados cubos épicos son divididos en stories más pequeñas.

4. Marco Referencial

4.1. Benchmarking

Como se mencionó en los beneficios, se pretende que la aplicación sea competitiva, y no hay mejor forma de hacerlo que tomando en cuenta aplicaciones consideradas en la tienda de Google como “Editor’s choice”.

Para describir el criterio “Editor’s choice” se tomará palabras de un post publicado en el Blog de Desarrolladores Android:

“Como parte de nuestros esfuerzos para reconocer este contenido en Play Store, hoy lanzamos Android Excellence. Las nuevas colecciones mostrarán aplicaciones y juegos que ofrecen increíbles experiencias de usuario en Android, utilizarán muchas de nuestras mejores prácticas y tienen un gran diseño, rendimiento técnico, localización y optimización de dispositivos.” Fahey, K. (2017)

Se toman en cuenta aplicaciones de la Play Store en la sección Editor’s choice debido a que el rediseño se basará en parte en la guía de diseño de Google. Sin embargo esto no quiere decir que el producto final fracasará a tener el mismo efecto en la tienda de Apple.

Para el benchmark se usaron características de las funcionalidades principales de obtención de itinerarios para movilidad en transporte público de las siguientes aplicaciones:

- Google Maps
- Moovit
- CityMapper

Características	AMB Mobilitat	Google Maps	Moovit	CittyMapper
En el buscador principal				
Buscador único	✗	✓	✓	✓
Filtrar las opciones de transporte que se muestran	✓	✗	✗	✓
Input por voz	✗	✓	✗	✗
Permitir definir favoritos (Hogar y Trabajo)	✗	✓	✓	✓
Flujo de búsqueda de itinerarios				
Historial de búsqueda	✓	✓	✓	✓
Mostrar itinerarios sin solicitar punto de partida (Selecciona automáticamente la ubicación del usuario)	✗	✓	✓	✓
Filtrar los itinerarios por tipo de transporte	✓	✓	✓	✗
Seleccionar el día y hora de partida o arribo al destino	✓	✓	✓	✓
Mostrar si hay alguna afectación en uno de los medios de transporte	✗	✓	✓	✗
Mostrar los tipos de transportes por itinerario	✓	✓	✓	✓
Mostrar si hay trasbordo	✓	✓	✓	✓
Mostrar nombre de la calle de partida	✗	✓	✓	✓

Mostrar opciones de transporte como biking, scooters, motos alquiladas, etc	✗	✓	✓	✓
En el itinerario seleccionado				
Detalle de cada sub recorrido (Nombre de las paradas del transporte, las líneas, cantidad de paradas, tiempo del sub recorrido, afectaciones si las hay)	✗	✓	✓	✓
Las indicaciones están agrupadas de acuerdo al transporte correspondiente (Ej. si se muestra recorrido en metro, el nombre de la línea y sus paradas son claramente identificables como relacionadas)	✓	✓	✗	✓
Muestra los horarios de partida y arribo de cada sub recorrido	✗	✓	✓	✓
Muestra el nombre de las calles de partida de cada recorrido	✗	✓	✓	✓
Inicio del recorrido				
Activación de un notificación persistente con indicaciones del sub recorrido actual	✗	✓	✓	✓
Indica en el mapa el tramo actual	✓	✓	✓	✓

Acompañamiento/indicaciones por audio según el sub recorrido	✗	✓	✗	✗
Opción de compartir ubicación en vivo	✗	✓	✗	✗
Buscador de líneas				
Buscador interno	✓	✗	✓	✗
Descarga de ficha informativa	✓	✗	✗	✗
Información contextual (Recorrido en texto y en mapa, horarios, afectaciones)	✓	✓	✓	✓
Filtrado por tipo de transporte	✗	✗	✓	✗
Alertas por líneas	✗	✗	✓	✗
Agregar a favoritos	✗	✗	✓	✓
Favoritos				
Agregar paradas	✓	✓	✓	✓
Agregar lugares personalizados (Cualquier punto en el mapa)	✗	✓	✓	✗
Lugares “Hogar” y “Trabajo” predefinidos	✗	✓	✓	✓
Agregar recorridos	✗	✗	✗	✓
Alertas afectación				
Información detallada por línea (descripción de la afectación y mapa)	✓	✗	✓	✗

Alertas personalizadas por línea de transporte específica	✗	✗	✓	✗
Otras funcionalidades				
Formulario de contacto/feedback mejoras	✓	✓	✓	✓
Cambio de idioma	✓	✗	✓	✗
Información sobre otros medios de transportes masivos	✓	✓	✓	✓
Configuraciones avanzadas (Mapas offline, cambio de unidad de distancia, notificaciones)	✗	✓	✓	✗

4.2. Procesos Relacionados

Al no tener contacto con la empresa AMB y por consecuencia no tener conocimiento sobre los procesos actuales de la aplicación, se proponen lo siguientes 3 procesos necesarios para el funcionamiento de la aplicación:

- **Procesos de operacionales de la organización**, son los encargados de generar la información contextual y operativa de los diferentes transportes públicos, para que los usuarios puedan usar los servicios de la aplicación.
- **Procesos de soporte de las aplicaciones (Android y iOS)**, son los encargados del mantenimiento de la aplicación (UI/UX y desarrollo) y recibir el feedback de parte de los usuarios.
- **Procesos de reportería/analítica**, son los encargados de medir los indicadores de rendimiento de la aplicación, además de ofrecer recomendaciones ante posibles problemas detectados (caídas de la aplicación, métricas bajas de uso de la aplicación, desinstalación, etc).

Una vez que se logre obtener información más concreta de los procesos actuales dentro de AMB, se podrá adaptar o mejorar los mismos, según el alcance del proyecto.

4.3. Casos de Uso

Actividades necesarias de los procesos operacionales de la organización:

- **Actualización y mantenimiento de itinerarios de líneas de transporte en tiempo real.** Es importante que la base de datos, de preferencia en la nube, con información de arribo por parada de los buses, trenes, metro y tranvía sea administrada y actualizada constantemente, ya que es uno de los pilares del servicio que ofrece la aplicación.
- **Actualización y mantenimiento de información general de líneas de transporte (paradas, recorrido, horarios por parada).** Esta base de datos, de preferencia en la nube, debe ofrecer a los usuarios de la aplicación información contextual de las diferentes líneas de transporte como por ejemplo: líneas que pasan en una parada, los horarios diversos según el día de la semana, recorrido en mapa y ficha informativa en formato pdf para descarga, etc. Este proceso es otro de los pilares fundamentales de la aplicación y por lo tanto debe tener un control igual de riguroso que la información de tiempo real.
- **Generación de alertas de alteraciones en servicios,** servicio que permite informar de manera detallada a los usuarios sobre problemas de movilidad y ofrecer alternativas. Esta información irá de la mano con las 2 anteriores actividades.

Actividades necesarias en el soporte de las aplicaciones (Android y iOS):

- **Soporte a los usuarios (sección de contacto),** recibir y gestionar las diferentes solicitudes generadas por los usuarios a través de la aplicación.
- **Control de bugs,** monitoreo del funcionamiento de la aplicación para generar informes de errores.
- **Soporte en las tiendas virtuales,** comprende la actualización de la información de las tiendas virtuales. Además de tener contacto con los usuarios, solucionar problemas y generar oportunidades de mejora para futuras versiones. Es un canal que sirve para determinar la satisfacción de los clientes con la aplicación.
- **Mantenimiento y actualización de la aplicación,** asegurarse que los diferentes servicios de backend se comuniquen correctamente por la aplicación. También toma las recomendaciones de las dos actividades anteriores para gestionar los cambios que sean necesarios.

Actividades necesarias en reportería analítica:

- **Seguimiento por medio de la plataforma Firebase**, control de las acciones realizadas por los usuarios y su comportamiento(analíticas) para poder tomar decisiones fundamentadas en relación con los objetivos de la empresa.
- **Log de caídas de la aplicación**, generar reportes de las caídas y las circunstancias específicas, para que el equipo de desarrollo pueda gestionar según la severidad del evento.

4.4. Stack Tecnológico

Para aprovechar al máximo las prestaciones de hardware y software de los diferentes dispositivos iOS y Android, se plantea un stack tecnológico nativo. Por lo tanto se usarán lenguajes de programación, kits/entornos de desarrollo y demás herramientas específicas que permitirán que el proyecto sea de fácil mantenimiento y escalable. A largo plazo implica un ahorro de mantenimiento y mejora aunque esto no quiere decir que tenga ciertos inconvenientes.

Los pros y contras del desarrollo nativo propuestas por son:

Pros	Contras
<ul style="list-style-type: none"> • Integración completa a dispositivos (cámaras, gps, micrófonos, etc) • Excelente experiencia de usuarios • Rápido performance • Adaptable a varios tamaños • Seguro 	<ul style="list-style-type: none"> • Funciona solo en una plataforma • Desarrollo más costoso

Para poder explicar mejor el stack, se lo dividirá en dos tecnologías:

- Tecnologías Client-side
- Tecnologías Server-side

4.4.1. Tecnologías Client-side

Se refiere a las diferentes tecnologías que se ejecutan en los dispositivos del usuario, la propuesta para el desarrollo Android es la siguiente:

- **Kotlin**, es un lenguaje de programación open-source funcional y/u orientado a objetos, diseñado por JetBrains basado en Java, es uno de los lenguajes más populares para el desarrollo de aplicaciones Android. Las ventajas de usar Kotlin son: código más simplificado y ligero, es compatible con dispositivos antiguos, es de rápida ejecución y genera APKs ligeras.
- **Android Developers Tools/Android SDK**, es un toolkit gratuito propio de Android que permite expandir el desarrollo de aplicaciones nativas. Ofrece soporte para código Android, herramientas de debugging, emuladores, etc.
- **Gradle**, es un toolkit gratuito específico para automatizar y administrar la creación de APKs, aunque también puede ser usado para aplicaciones iOS.

Para el trabajo de desarrollo en dispositivos iOS se necesitarán las siguientes tecnologías:

- **Swift**, es el lenguaje de programación más popular para iOS, basado en Objective-C. Es más práctico debido a su sintaxis simple y menos propensa a errores. Contiene librerías que optimizan los recursos y como resultado la aplicación final es más ligera de peso y de rápido performance.
- **iOS SDK**, es el kit de desarrollo de software gratuito para iOS, permite acceder a varias funcionalidades y servicios de hardware o software de los móviles iOS. Además contiene un emulador que permite hacer pruebas a los desarrolladores sin la necesidad de tener un dispositivo a mano.

4.4.2. Tecnologías Server-side

Se refiere a las diferentes tecnologías, generalmente ubicadas en la nube, que administran las diferentes solicitudes realizadas por los usuarios. La propuesta de backend para soporte de aplicaciones Android y iOS es la siguiente:

- **Nginx**, es un servidor HTTP de alto rendimiento gratuito, en este proyecto se lo propone usar como reverse proxy, esto permitirá balancear la carga entre diferentes servidores, ofrecer mejor seguridad, incrementar el rendimiento entre las diferentes solicitudes (de cliente o dinámicas) y aplicar auditorías fácilmente.
- **Node.js**, es un entorno de código abierto del lado del servidor que permite ejecutar código escrito en JavaScript, puede desplegarse en diferentes plataformas (Windows, Linux, Mac OS, etc). Dado el hecho de que se propone trabajar con Kotlin, es clave crear una aplicación Javascript end-to-end. Gracias a

esto la aplicación tendrá mejor rendimiento, facilitará el mantenimiento y la escalabilidad.

- **Express**, es uno de los frameworks gratuito de servidores más populares para web, en lo que se refiere al uso de Node.js. Es un framework mínimo y flexible que provee herramientas robustas para aplicativos webs o móviles.
- **MongoDb**, es una base de datos orientada en documentos open source, que almacena datos en documentos de estilo JSON. Permite administrar rápidamente grandes cantidades de datos y está optimizado para infraestructuras sociales o de aplicativos móviles, además cuenta una gran comunidad activa que provee soporte continuo.
- **Mapbox API**, es un servicio popular gratuito/pago que permite la implementación de mapas para aplicaciones web o aplicación. Entre las funcionalidades ofrece: mapas personalizados, localización, rutas, búsqueda de ubicaciones y navegación a partir de los datos de OpenStreetMap. Seleccionado inicialmente por encima de Google Maps, ya que no cobra antes de las 50.000 solicitudes mensuales (Google Maps cobra después de 20,000, a la fecha de creación de este documento) y es más barato por cada 1,000 solicitudes extras en comparación a la API de Google Maps (\$0.50 por solicitud extra a diferencia de \$1.00-14.00). Para la etapa de desarrollo este servicio no incurrirá en costos pero según los requerimientos finales del cliente se podría cambiar la API de mapas.

4.5. Datos

4.5.1. Leyes Aplicables

Se propone la recolección de datos y por tratarse de una aplicación que será usada dentro de espacio de la UE, se debe cumplir con el dictamen europeo 2/2013 artículo 29⁸, el reglamento UE 2016/679⁹ sobre las aplicaciones de los dispositivos inteligentes. Entre los puntos claves que se deben cumplir están:

- **Política de privacidad, finalidad y calidad de datos**, la aplicación debe estar definida en su totalidad. Se debe establecer la política de privacidad de la empresa, señalar los datos que se van a recabar/almacenar, la razón de recabar los datos y el tiempo que estos datos serán almacenados. Hay que tomar en

⁸ Opinion 03/2013 on purpose limitation https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2013/wp203_en.pdf

⁹ Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo de 27 de abril de 2016 <https://www.boe.es/doue/2016/119/L00001-00088.pdf>

cuenta que solo se deben recabar los datos necesarios para poder prestar el servicio de la aplicación.

- **Consentimiento**, ya que no es permitido el consentimiento tácito ni el consentimiento por tiempo transcurrido sin oposición, es necesario que de parte del usuario el consentimiento sea libre, informado, específico e inequívoco.
- **Los derechos ARCO** (aceptación, rectificación, cancelación y oposición), son extendidos al derecho al olvido y la portabilidad de los datos, se debe establecer claramente cómo se pueden ejercer estos derechos.
- **Control de brechas de seguridad**, ante cualquier vulnerabilidad de datos detectada se debe informar a los usuarios en un plazo máximo de 72 horas.

4.5.2. Recolección de Datos

La recolección de datos es importante para conocer el comportamiento de los usuarios, estar al tanto de sus necesidades, deseos y su actitud hacia el producto. A futuro por medio de reportería y analítica, permitirá tomar decisiones informadas para enfocar mejor los esfuerzos.

Por lo tanto, los datos de los usuarios que se propone recolectar de manera anónima, gracias a la aplicación del concepto de *privacidad por diseño*¹⁰, son:

- Email (para crear cuenta y sincronizar datos en otros dispositivos)
- Historial de búsquedas
- Líneas de transporte, paradas, locaciones y rutas más usadas (por búsquedas, descargas de fichas o añadidas al apartado de favoritos).
- Líneas de transporte más buscadas (ficha informativas)

Los datos de los usuarios que no se captarán son:

- Ubicación en tiempo real
- Contraseñas

Además se propone usar las herramientas de Firebase como Google Analytics, para poder capturar datos derivados de otras acciones como:

- Datos en tiempo real
- Idioma más popular

¹⁰ Data protection by design and by default <https://gdpr-info.eu/art-25-gdpr/>

- Cantidad de usuarios
- Cantidad de descargas de la aplicación
- Reporte de caídas/bugs
- Tiempo de uso/sesiones
- Secciones de salida en la aplicación
- Flujos de navegación
- Acciones/pantallas más populares

4.5.3. Explotación

Entre los beneficios específicos que conlleva recolectar los datos anteriormente mencionados están:

- **Acciones de marketing digital**, conocer a los usuarios y su accionar dentro de la aplicación se pueden dirigir campañas segmentadas para promover la imagen de la marca, promocionar nuevos servicios, etc.
- **Acciones de diseño UI/UX**, al conocer en qué sección de la aplicación salen los usuarios, se pueden tomar acciones de diseño basadas en datos(ej. mapas de calor) como por ejemplo cambiar la estructura/contenido de una sección para contrarrestar los valores negativos que se hayan detectado.
- **Control de proyectos en tiempo real**, no solo basta tomar acciones sino monitorear sus efectos. Al poder reconocer tendencias negativas en las métricas, se podrán tomar acciones instantáneas para contrarrestarlas y evitar afectaciones serias que influyan a una imagen negativa del producto.

4.6. Riesgos del Proyecto

4.6.1. Riesgos Tecnológicos

Como se mencionó anteriormente, los riesgos posibles para los proyectos digitales de esta naturaleza son los siguientes:

- No poder implementar funcionalidades planteadas en el prototipo por problemas de compatibilidad con el sistema de entrega de información actual (software o hardware) o que los cambios/mantenimientos de estos sistemas informáticos tengan un coste más alto del esperado.
- Que unos de los elementos del stack tecnológico deje de tener soporte y la migración resulte en un proceso complejo y costoso.
- Tener que depender de terceros para la prestación de servicios indispensables y no tener el control sobre los mismos.

- Tener que crear diferentes versiones de la aplicación debido a que ciertas funcionalidades no son compatibles con dispositivos de gama baja.
- Que el cliente no de feedback consistentemente
- Al usar Material Design para iOS, se corre el riesgo que nuevas funcionalidades propias de iOS no tengan su versión actualizada o no puedan ser implementadas en Material Design.

4.6.2. Riesgos Económicos

- Que los usuarios no se interesen por usar la aplicación y opten por aplicaciones ya conocidas más populares y la inversión no logre alcanzar los beneficios planteados anteriormente.
- Que la aplicación sea muy popular entre los usuarios actuales del transporte público y los costos de servicios de la API de mapas incrementen, afectando el presupuesto.

4.6.3. Riesgos de la Metodología

Entre los riesgos posibles de las metodologías Agile podemos encontrar:

- No contar con el dueño del producto para la toma de decisiones durante los sprints de descubrimiento.
- Que el cliente desee agregar nuevas funcionalidades al backlog o existan cambios constantes en prioridades del backlog, afectando los tiempos de entrega/presupuesto.
- Estimación de tiempos muy optimista.
- Cambios en el equipo por razones externas
- Cambios o definición no apropiada de criterios de aceptación del sprint
- Aprendizaje de Skills nuevos
- Dinámicas negativas en el equipo de trabajo

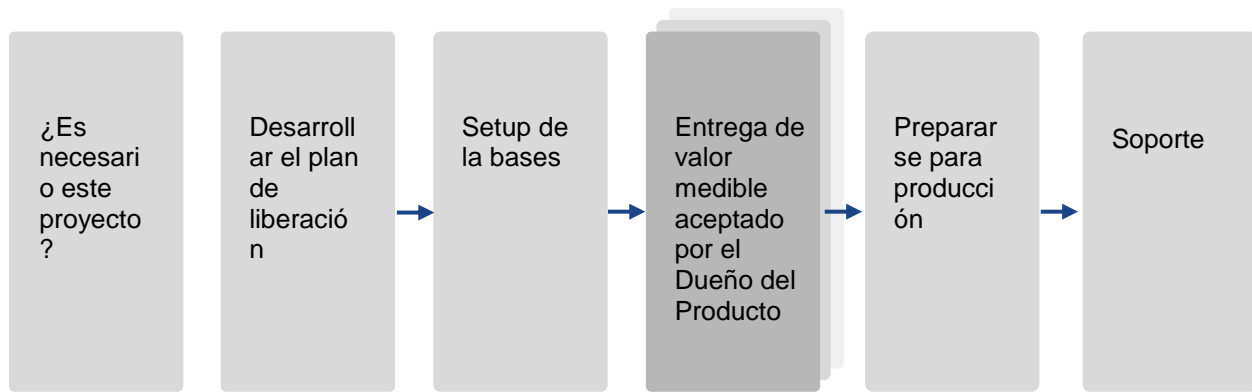
5. Planificación

5.1. Cronograma

ETAPA	TAREAS	APP MOBILITAT																																			
		MES 1				MES 2				MES 3				MES 4				MES 5				MES 6				MES 7				MES 8				MES 9			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34		
1	Definición del proyecto																																				
1.1	Análisis del proyecto (Viabilidad, VOI, Alternativas)																																				
1.2	Análisis de infraestructura tecnológica actual																																				
1.3	Definición alto nivel de features de la app/Roadmap																																				
1.4	Definición de perfiles de equipo de trabajo																																				
1.5	Estimación de presupuesto																																				
2	Planeación de liberación																																				
2.1	Creación del backlog																																				
2.2	Priorizar stories/asignar puntos de complejidad																																				
2.3	Desarrollo del plan de liberación (Cronograma más detallado)																																				
2.4	Actualizar presupuesto																																				
3	Sprint 0, Setup																																				
3.1	Setup de infraestructura																																				
3.2	Setup de tecnología (desarrollo y diseño)																																				
3.3	Modelos de arquitectura de app																																				
3.4	Actualización del modelo de la DB																																				
3.5	Look and feel y navegación																																				
3.6	Entrenamiento DualTrack y Design Sprint																																				
4	Ejecución de sprints de descubrimiento y desarrollo (Android/iOS/Backend)																																				
4.1-1	Sprint Diseño Feature 1																																				
4.1-2	Sprint Diseño Feature 2																																				
4.1-3	Sprint Diseño Feature 3																																				
4.1-4	Sprint Diseño Feature 4																																				
4.1-5	Sprint Diseño Info de Contexto																																				
4.2-1	Sprint Desarrollo Feature 1																																				
4.2-2	Sprint Desarrollo Feature 2																																				
4.2-3	Sprint Desarrollo Feature 3																																				
4.2-4	Sprint Desarrollo Feature 4																																				
4.2-5	Sprint Desarrollo Info de Contexto																																				
5	Pre-entrega																																				
5.1	Testeo final de la app																																				
5.2	Documentación frontend y backend																																				
5.3	Entrenar staff de producción de backend																																				
5.4	Salida a producción																																				
6	Producción																																				
6.1	Gestionar el sistema backend																																				
6.2	Soporte al sistema backend/Entrega de documentación																																				
6.3	Detectar problemas y mejoras																																				
6.4	Transición al equipo de soporte de AMB																																				
6.5	Campaña de Marketing																																				

5.2. Plan de Trabajo

Como se propone llevar a cabo el proyecto mediante metodología Agile, lo correcto es plantear una plan acorde con los hitos y actividades correspondientes. En un nivel alto se propone el siguiente ciclo para el proyecto:



Ciclo de vida de un proyecto Agile

- **Iniciación**, etapa donde se analiza si el proyecto es viable o necesita cambios:
 - Visión general del proyecto
 - Justificar el VOI
 - Analizar alternativas
 - Identificar los atributos del equipo de trabajo
 - Definir funcionalidades de nivel alto
 - Crear del roadmap
 - Crear presupuesto de estimación de costos
 - Obtener de soporte tecnológico
- **Planeación de liberación**, considerado como el sprint -1 donde se define un poco más a detalle el proyecto y lo que se debe llevar a cabo:
 - Crear el backlog de funcionalidades
 - Priorizar los stories basado en los valores que aporta
 - Estimar los puntos de los stories
 - Desarrollar el plan de liberación (cronograma más detallado)
 - Actualizar del presupuesto de estimación de costos
- **Sprint 0**, o sprint de preparación donde se pone a punto el equipo de trabajo y todo lo necesario para comenzar con los sprints de entrega:
 - Setup del ambiente de desarrollo físico y tecnológico

- Crear modelos de arquitectura de tecnología de alto nivel
- Actualizar el modelo de base de datos de alto nivel
- Look and feel y navegación
- Entrenar el equipo de trabajo en procesos ágiles
- **Ejecución de los sprints**, etapa donde se pondrán en marcha los sprints de descubrimiento (Design Sprint) y desarrollo (Scrum):
 - Descubrimiento
 - Definir los problemas a resolver
 - Definir soluciones
 - Priorizar soluciones
 - Crear storyboard
 - Creación de prototipo baja calidad
 - Test de usuarios
 - Tomar feedback e iterar el prototipo
 - Crear el prototipo de alta calidad
 - Entregar el prototipo validado al camino de desarrollo
 - Actualizar el Sistema de Diseño
 - Desarrollo
 - Reunión de planeación de sprint
 - Reuniones diarias
 - Cambios de backlog
 - Tests de QA
 - Revisar código fuente
 - Entregar el sprint y aprobar el trabajo
 - Reunión de retrospectiva
 - Monitorear indicadores de rendimiento/velocidad
 - Actualizar el Sistema de Diseño
- **Pre-Entrega**, etapa de validaciones finales y finalización de documentación de soporte:
 - Testeo final de la aplicación
 - Finalizar documentación frontend y backend
 - Entrenar staff de producción de backend
 - Salida a producción en las tiendas virtuales de Google y iOS

- **Producción**, etapa donde la aplicación ya está liberada a los usuarios finales, se le da soporte al sistema, se mide el rendimiento y se comunica su liberación por medio de marketing:
 - Gestionar el sistema backend
 - Dar soporte al sistema backend
 - Identificar problemas y mejoras
 - Iniciar transición al equipo de soporte de AMB
 - Iniciar campaña de marketing online y offline

5.3. Recursos Necesarios

5.3.1. Recursos Humanos

Es recomendable trabajar con un equipo pequeño de profesionales, en metodologías Agile, por lo tanto es muy importante que los involucrados en el equipo de trabajo sean autosuficientes y puedan cumplir cualquier tarea necesaria en lo posible. Los perfiles profesionales necesarios para el proyecto son los siguientes:

- **1 dueño del producto**, es uno de las personas más importantes dentro del proyecto ya que es el proxy entre el cliente y la empresa, además de dar el visto bueno de las diferentes entregas. Es un rol imprescindible del proceso de Scrum.
- **1 ScrumMaster**, otro rol necesario para poder utilizar Dual-Track Scrum. Profesional que entiende las necesidades del cliente tanto como las del equipo de trabajo, conoce de tecnología y negocios. Es quién guía al equipo y su ritmo de trabajo, es otro rol primordial.
- **2 diseñadores UI/UX senior (Android y iOS)**, la necesidad de contar con dos profesionales de diseño app seniors, recae en el trabajo para dos ecosistemas diferentes, con ciertas diferencias en interacciones, necesario para usarlas correctamente y ofrecer una experiencia nativa según lo permita Material Design. Se podrá sacar provecho de la experiencia y creatividad de dos seniors al momento de las reuniones del Design Sprint. Serán encargados de crear el sistema de diseño, facilitando el desarrollo durante la etapa de creación y dejando documentación para cambios futuros.
- **1 desarrollador android/iOS full stack senior**, se requiere este profesional ya que el proyecto, en la propuesta actual, requiere cambios significativos de backend. Será quien pueda asesorar en el camino de descubrimiento sobre las mejores decisiones para mantener el proyecto dentro del presupuesto y cronograma. También será muy importante en la implementación de APIs, métodos y base de datos para recuperar información para el uso dentro de la aplicación.

- **2 desarrolladores android/iOS full stack junior**, perfiles junior pero multidisciplinarios que puedan cumplir tareas de desarrollo sin importar la plataforma. Se necesitará 1 persona para cada sistema operativo.
- **1 Quality Assurance**, profesional encargado de revisión de calidad del producto (backend y frontend) que trabajará junto a los desarrolladores. Al ser un proceso tecnológico, es de esperarse bugs que los desarrolladores no hayan descubierto, es necesario tener alguien enfocado en encontrar vulnerabilidades en el sistema para obtener buenos resultados en las reuniones de entrega. Este perfil estará encargado de crear un manual de uso para el cliente.

5.3.2. Recursos Materiales no Tecnológicos

Los recursos materiales básico son los siguientes:

- **Coworking**, en Dual-Track Scrum es ideal que el equipo comparta un espacio físico para poder realizar las diferentes ceremonias (sprints de descubrimiento con el cliente y usuarios, planeación de sprints, reuniones diarias, retrospectiva del sprint, etc).
- **Materiales de oficina**, aunque se trate de un proyecto tecnológico y todo lo acordado quede registrado en la nube, los procesos de toma de decisiones o creativos en su mayoría, por agilidad, se desarrollan en medios físicos.

5.3.3. Recursos de Hardware

Los recursos de hardware para el trabajo de los diferentes perfiles profesionales son:

- **5 portátiles 17 pulgadas con Windows OS**, 2 para roles administrativos, 1 para QA y 2 para los diseñadores. Se requieren de portátiles por la facilidad de transporte.
- **1 Macbook pro 15 pulgadas**, 1 para el desarrollador senior que necesitará validar las aplicaciones en sus diferentes sistemas operativos y hay que tomar en cuenta que el desarrollo en iOS está limitado a sistemas Mac OS.
- **2 iMacs 27 pulgadas**, para el equipo de desarrollo junior se requieren iMacs para la aplicación iOS, vale mencionar que la versión Android puede ser trabajado sin problemas en sistemas Apple.
- **3 monitores 24 pulgadas widescreen+ratones+teclados**, al tener monitores de estas características, permitirá mayor flexibilidad y como resultado mayor eficacia al equipo de trabajo. Se beneficiarán el desarrollador senior y los

diseñadores que a medida que avanza el proyecto su necesidad de visualización de artboards incrementa de igual medida.

5.3.4. Recursos Software

Antes de explicar el motivo de esta selección, cabe mencionar que algunos de los softwares tienen versiones gratuitas y de pago. En esta propuesta se está recomendando trabajar con las versiones de pago (costos incluidos en el presupuesto) ya que hay más funcionalidades disponibles.

Solo la API para la visualización de mapas, se recomienda usarla durante el proceso de desarrollo en versión gratuita. Si los requerimientos cambian se podría analizar el cambio de API como la de Google Maps.

- **Administración y comunicación general**
 - **Google Docs/Slides/Keep/Sheets/Drive**, herramientas gratuitas cloudbase para la documentación de administración, colaboración y para la elaboración de los manuales del sistema.
 - **Jira+Confluence**, software gratuito/pago cloudbase para la gestión, administración, tracking del proyecto a nivel tecnológico y comunicación interna.
 - **Slack**, plataforma gratuita/pago para la comunicación interna y externa
- **Equipo de diseño+UX**
 - **Figma**, herramienta gratuita/pago cloudbase para el diseño de los prototipos de baja y alta calidad.
 - **Morae Techsmith**, software de pago indispensable para los test de usuarios y poder generar rápidamente informes durante los sprints de descubrimiento.
- **Equipo de desarrollo+QA+Integración continua**
 - **Android Studio**, es el Ambiente de Desarrollo Integrado (IDE en inglés) oficial y gratuito para aplicaciones Android. Ofrece herramientas para editar, testear y ejecutar código fuente.
 - **Xcode**, software gratuito necesario para la creación de aplicaciones basadas en iOS.
 - **Github**, repositorio cloudbase gratuito/pago para respaldo seguro de los avances del equipo de desarrollo.

- **Bitrise**, herramienta gratuita/pago para la integración/entrega continua, además permite a los desarrolladores y QA testear la aplicación en diferentes dispositivos virtuales.

5.4. Indicadores de Rendimiento

La metodología Dual-Track Scrum, nos ayudará con la evaluación constante de los KPIs del proyecto a través de métricas específicas.

Sealights¹¹, establece que las métricas Agile ayudan a medir el proceso de desarrollo, productividad, calidad del trabajo, predictibilidad, estado del equipo de trabajo y del producto elaborado. No se enfocan en medir el “qué” o “cuánto” del proyecto sino en cómo el proyecto impacta al cliente.

Sealight define que las métricas Agile tienen 3 objetivos claros:

- Medir entregables
- Medir efectividad
- Medir al equipo

5.4.1. Métrica de Entregables

Sirven para medir el trabajo realizado por el equipo de trabajo y entender cuánto valor se está entregando a los clientes:

- **Prototipos validados (camino de descubrimiento)**, son los prototipos con funcionalidades específicas, que han sido puestos a prueba con los usuarios finales y han sido refinados en los sprints de iteración.
- **Éxito del sprint (camino de desarrollo)**, se define conociendo si en el sprint se han alcanzado a realizar todas las tareas propuestas en el backlog del sprint terminado.
- **Defectos escapados y densidad de los defectos (camino de desarrollo/descubrimiento)**, permite conocer cuántos bugs se encuentran en la labor del profesional de QA o en un test de usuarios. La densidad permite conocer cuántos errores se encuentran por cantidad de líneas de código.

¹¹10 Powerful Agile Metrics - and 1 Missing Metric <https://www.sealights.io/software-development-metrics/10-powerful-agile-metrics-and-1-missing-metric/>

- **Velocidad del equipo (camino de desarrollo)**, se mide observando en promedio cuántas stories del backlog del sprint fueron completadas. Permite estimar cuánto trabajo puede ser completado a futuro por el equipo.
- **Sprint Burndown (camino de desarrollo)**, es un gráfico que representa el progreso de un sprint o del proyecto, a través de las horas o puntos restantes.

5.4.2. Métricas de Efectividad

Sirven para medir el esfuerzo del equipo de trabajo, su contribución al negocio en términos de VOI, tiempo de comercialización, etc:

- **Tiempo de comercialización**, tiempo en el cual el producto comienza a dar valor a los usuarios o el tiempo en que demora en generar ingresos.
- **Satisfacción del cliente**, se calcula a través de encuestas a usuarios, o en caso de aplicaciones observando las calificaciones y comentarios de las tiendas virtuales.

5.4.3. Métricas de Equipo de Trabajo

Sirven para medir al equipo de trabajo para conocer la “salud” del mismo y encontrar problemas como desgaste, insatisfacción o rotación del equipo:

1. **Scrum diario**, se mide a través de las conclusiones documentadas, provee feedback cualitativo del progreso del equipo y su “salud”.
2. **Rotación del equipo**, se calcula viendo cuántos integrantes han ido cambiando a través de los sprints.

6. Presupuesto

La estimación de costos para el proyecto es la siguiente:

Concepto	Tiempo/Meses	Costo/Mes	Importe (€)
RRHH			
ScrumMaster/Project Manager	9	2.500,00 €	22.500,00 €
Dueño del Producto	7	1.300,00 €	9.100,00 €
Desarrollador android/iOS full stack senior	6	1.800,00 €	10.800,00 €

Desarrollador android/iOS full stack junior (x2)	10	1.200,00 €	24.000,00 €
Diseñador UI/UX Android y iOS senior (x2)	10	1.400,00 €	14.000,00 €
Quality Assurance	5	1.300,00 €	6.500,00 €
			86.900,00 €
Profesionales Externos			
Agencia de Marketing	2	4.000,00 €	8.000,00 €
			8.000,00 €
Administración			
Coworking	9	550,00 €	4.950,00 €
Material de oficina	9	230,00 €	2.070,00 €
			7.020,00 €
Software/Hardware			
Servidor de pruebas	9	250,00 €	2.250,00 €
Software diseño y desarrollo (Sin Maps API)	7	90,00 €	630,00 €
			2.880,00 €
Total+Impuestos=			129.952,00 €

7. Consideraciones

7.1. Recomendaciones Iniciación y Setup

Es importante decidir si se realiza lo propuesto en este documento o si se hace una integración al sistema actual de AMB. En especial si los métodos que alimentan la aplicación actual no son tan completos como los propuestos en el prototipo, se necesitará realizar un trabajo extenso en la parte del backend.

Hay que tomar en cuenta que la API de mapas es parte fundamental del servicio de la aplicación, por lo tanto es importante decidir si se continúa con la API actual (Google

Maps¹²) o se cambia a Mapbox¹³. Durante el proceso de desarrollo, no importa que API se seleccione, ninguna causará gastos, esto solo ocurrirá al momento de entrada a producción. No hay duda que Google Maps es la plataforma de mapas más robusta del mercado, pero los costos pueden sumar rápidamente si se tiene una gran base de usuarios.

7.2. Consideraciones Sprints de Descubrimiento

7.2.1. Design Sprint

Antes de comenzar en el sprint es necesario elaborar *User Personas*¹⁴ básico, no muy detallado. De esta manera los involucrados tendrán claro para quién se está elaborando el producto y también será fácil identificar personas para los test de usuarios.

Para las dos primeros días que podrían considerarse de planeación, es sumamente importante que los stakeholders o un delegado de los mismo esté presente ya que el trabajo de los días restante se basará en la soluciones tomadas mediante procesos flexibles como *Lightning Decision Jam*¹⁵.

Al momento de realizar los test de usabilidad, los prototipos serán de baja o media calidad, es más importante el valor que aporta la solución/funcionalidad que lo agradable que pueda ser la interfaz. Solo cuando se tenga un feedback positivo por parte de los usuarios y los stakeholders, se puede proceder a realizar un prototipo de alta calidad para entregar a camino de desarrollo.

7.2.2. Google Material Design

Material Design es un Sistema de Diseño creado por Google, en un artículo reciente podemos encontrar la siguiente definición de lo que es un Sistema de Diseño:

"...son esencialmente colecciones de reglas, limitaciones y principios, implementados en diseño y código. Estos tres atributos sirven funciones diferentes y proveen un orden sistemático y coherente en sistemas desde botones hasta páginas completas." (Sylvee, 2018)

¹² <https://cloud.google.com/maps-platform/pricing/>

¹³ <https://www.mapbox.com/pricing/>

¹⁴ <https://www.usability.gov/how-to-and-tools/methods/personas.html>

¹⁵ <https://uxplanet.org/lightning-decision-jam-a-workshop-to-solve-any-problem-65bb42af41dc>

Material Design es un framework para trabajar proyectos digitales que permite crear y escalar el diseño de una web o aplicación, de forma coherente manteniendo la uniformidad. El Sistema de Diseño para la aplicación de AMB, irá tomando forma a medida que se vayan superando sprints de descubrimiento gracias a los siguientes entregables:

- **Guía de estilo**, enfocado en los estilos gráficos (ilustraciones, iconografía, paleta de colores, tipografías, etc) y su uso correcto según el contexto de aplicación.
- **Biblioteca de patrones**, la recolección de todos los elementos de la interfaz gráfica (cajas de texto, botones, menús, etc.) exponiendo sus características, sus reglas de uso y el código de programación necesario para su implementación (trabajado por el grupo de desarrollo).

Entre los beneficios de los sistemas de diseño están:

- Escalar en equipos de diseño sin importar si son grandes o pequeños, todos trabajarán sobre un lenguaje de diseño común y así diferentes secciones tendrán una imagen relacionada.
- Administrar la “deuda de diseño”, al tener elementos que se comportan y lucen de cierta manera su implementación en desarrollo, a futuro ahorra tiempo y dinero.
- Diseñar consistentemente y lograr que los usuarios se familiaricen con el producto. Al mismo tiempo permitir a los diseñadores enfocarse en la experiencia del usuario y no en el diseño como tal.
- Prototipar rápidamente debido a que los elementos ya existen y solo son usados como piezas de lego.
- Iterar rápidamente, lo cual implica un ahorro de esfuerzos en el desarrollo cuando se realizan cambios en el diseño.
- Mejorar la usabilidad al ofrecer una interfaz coherente y reducir la carga cognitiva de los usuarios.
- Aplicar reglas de accesibilidad

Se usará de base Google Material Design con el objetivo de crear un sistema propio de la aplicación. La razón principal es la experiencia y conocimiento en la creación de productos digitales con los que cuenta Google. Otra razón para el uso de esta guía, es que va de la mano con una guía de desarrollo open source para iOS y Android,¹⁶¹⁷ y

¹⁶Build a Material Theme <https://material.io/tools/build-a-material-theme/#>

como beneficio permitirá a los desarrolladores enfocar esfuerzos en tareas más críticas del proyecto como las de backend.

El uso de esta guía no quiere decir que el producto propuesto vaya a ser genérico o parecerse a una aplicación de Google, Material Design es sumamente flexible en diseño y desarrollo, hay muchos casos de estudios¹⁸ en los que se puede apreciar el uso del sistema sin caer en la homogeneidad. Esto permitirá diseñar la aplicación para iOS y Android con eficiencia y eficacia durante el paso de diseño a desarrollo.

7.2.3. UI, UX y Prototipo

Regresando a los beneficios del Material Design, permitirá dar valor agregado al diseño, poniendo en uso buenas prácticas como micro interacciones y accesibilidad, potenciando la UX de toda la aplicación.

Para poder sacarle el provecho máximo, es necesario que los diseñadores trabajen con herramientas que permitan la creación de un Sistema de Diseño, como Figma. Figma permite automatizar esta tarea mientras se realizan tareas normales de diseño y facilita la colaboración entre profesionales de todo el equipo de trabajo, gracias a su naturaleza cloud base.

Aunque se trabaje bajo una guía Android los diseñadores deberán crear una librería de patrones híbrida (en componentes que lo ameriten) para evitar incongruencias de UI al momento de desarrollar las versiones iOS y Android.

Como se mencionó en el apartado de *Estado Actual del Proyecto*, los test de usabilidad hallaron oportunidades de mejoras, las cuales fueron implementadas en un prototipo funcional de alta calidad, versión Android (Anexo 1). Este prototipo permitirá tener a los diseñadores una base para trabajar y poder generar varias propuestas de look and feel en la etapa de setup de sprints para la versión iOS y Android. Esto beneficiará a los primeros sprints de descubrimiento, al poder entregar los prototipos rápidamente.

7.3. Consideraciones Sprints de Desarrollo

Los desarrolladores deberán trabajar conjuntamente con los diseñadores para evitar problemas de comunicación en el producto final y en la librería de patrones del Sistema

¹⁷ Documentación de desarrollo de Material Design <https://material.io/develop/>

¹⁸ Acerca nuestros estudios de Material Design <https://material.io/design/material-studies/>

de Diseño. Cada desarrollador estará encargado en crear y actualizar código dentro de esta librería. Cada variación para el sistema operativo específica deberá estar documentada para referencia durante el proyecto y cambios futuros.

8. Política de Difusión

Para dar a conocer la nueva versión de la aplicación se propone una campaña de marketing online y offline, su objetivo, estrategias y plan de acción serán descritos en los siguientes puntos.

8.1. Objetivo

El objetivo principal de la campaña es dar a conocer la nueva aplicación con sus nuevas funcionalidades a los usuarios objetivos y a los potenciales, además de crear una nueva imagen de marca de la aplicación.

Es importante recordar que las calificaciones de las tiendas Play Store y App Store no son ideales (por debajo de las 4 estrellas), por lo que se recomienda potenciar la campaña, para que los usuarios que ya tienen la aplicación instalada le den una nueva oportunidad y a los que no la tengan, crearle el interés por probarla.

8.2. Estrategia

La campaña tendrá dos canales de difusión, el principal será el canal online y como soporte adicional se tendrá al canal offline.

Las estrategias propuestas para la campaña online:

- **Marketing de contenidos en redes sociales**, por medio de videos animados tipo tutorial, mini spots con influencers, post gráficos/animados informativos para generar interés del público objetivo y potencial.
- **Promoción de pago en Adwords**

Las estrategias para la campaña offline son:

- **Publicidad BTL**, en paradas de transporte y puntos turísticos más importantes
- **Publicidad ATL**, envío de un boletín de prensa para los diferentes medios de comunicación.

8.3. Plan de acción

El plan de acción dura 1 mes, estará precedido por tareas de planeación, creación de identidad de la campaña y elaboración de piezas gráficas. Durante el segundo mes se propone el siguiente cronograma:

	Online Marketing		Offline Marketing	
Tipo	Redes Sociales	Adwords	BTL	ATL
Semana 1	<ul style="list-style-type: none">Comunicados en la web de AMBLanding nuevo en web de la AMB	<ul style="list-style-type: none">Publicidad continua	<ul style="list-style-type: none">Campaña de afiches en paradas y busesCampaña de afiches y spots en aeropuerto	<ul style="list-style-type: none">Envío de boletines a medios de comunicación de lanzamiento de nueva aplicación
Semana 2	<ul style="list-style-type: none">Campaña de contenido en redes sociales (Instagram y Twitter)			
Semana 3				
Semana 4				
Indicador de éxito	<ul style="list-style-type: none">Comentarios positivos en redes sociales y tiendas de las aplicacionesMonitoreo de analíticas de las diferentes redesTráfico a la tienda procedente de campaña (diferencia con el tráfico del periodo anterior)Descargas de la aplicación en Play Store y App Store		<ul style="list-style-type: none">Tráfico procedente de campaña, por medio de códigos QR	<ul style="list-style-type: none">Cantidad de menciones en medios de comunicación

9. Fuentes Documentales Consultadas

Fahey, K. (2017). Recognizing Android Excellence on Google Play. Recuperado 13 abril, 2019, de <https://android-developers.googleblog.com/2017/06/recognizing-android-excellence-on.html>

Scrum.org. (2010). Scrum Guide. Recuperado 13 abril, 2019, de <https://www.scrumguides.org/scrum-guide.html>

Sylvee, L. (2018, 28 septiembre). What is a Design System?. Recuperado 17 abril, 2019, de <https://medium.muz.li/what-is-a-design-system-1e43d19e7696?gi=dc46f9b78427>

Usability.org. (2004). Usability Testing | Usability.gov. Recuperado 13 abril, 2019, de <https://www.usability.gov/how-to-and-tools/methods/usability-testing.html>

10. Anexos

Anexo 1. Prototipo de mejoras planteadas (Archivo .pdf 3,092 KB)